International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 3, No.3, July 2012 – www.ijais.org

Automatic Data Migration between Two Databases with Different Structure

Ramin Ahmadi
Faculty of Computer & IT Engineering
Mazandaran University of Science & Technology
Babol, Iran

Bagher Rahimpour Cami
Faculty of Computer & IT Engineering
Mazandaran University of Science & Technology
Babol, Iran

Hamid Hassanpour
Faculty of Computer & IT Engineering
Shahrood University of Technology
Shahrood, Iran

ABSTRACT
The process of data migration is one of the major challenges in the database structure development and improvement. The huge size of databases and the difference in syntax and semantics of tables are some of the crucial problems in data migration field. Data migration is performed in two phases sequentially: structure mapping and migration. In traditional approach, the first phase is carried out by the use of human experts, and the second phase is done by a software. In new data migrating approaches, the process is carried out automatically by the use of ontology in the first phase. In this paper with studying various methods in this field, a new model for migrating data between two databases with a different structure is proposed. By reviewing the existing models for traditional and semantic databases, the proposed model focuses on data migration between two traditional and semantic databases. In this model, data migration process is carried out by creating a middle layer. Covering both semantic and traditional databases is one of the most important features of proposed model.

Keywords
Data Migration, Semantic Data Migration, Semantic Database, Ontology.

1. INTRODUCTION
Data migration is the process of migrating data between two database management systems [1]. Generally in data migration, the data is migrated from an old platform to a new one. Database administrators use data migration to fulfill the database needs and updating its structure.

The data migration-which involves extracting data from source database and loading the data to the destination database-, faces serious challenges. Rapid growth of database on one hand and the structural differences on the other hand, are the main difficulties of data migration. The structural difference of databases may be due to semantic or syntax.

The majority of proposed approaches emphasize on data migration based on the structural difference of the databases [2]. The difference between two databases can be classified in three categories:

a. Syntax Difference

This problem originates from the usage of variant languages in two databases. In this case, an extensive knowledge base is needed to cover the grammar of both languages. For data migration, the request should be written and sent in source database language and also output data should be stored by the use of a target database language.

b. Data Model Difference

This problem originates from the use of different data models such as object oriented or relational, in the databases. In case of the aforementioned situation, the data model of the source database should be mapped and its equivalent should be implemented in the data model of target database. In this situation, using a software layer such as a wrapper for converting the two structures is an appropriate solution.

c. Semantic Difference

Variant analysis and aspect toward the process of developing a database causes semantic differences between two databases. Due to this inconsistency, different entities are created for a certain issue. For instance, consider the interconnected academic systems of two universities in Australia and United States. These two need to share educational data for research purposes [3]. Even if the whole data is in English and the two universities use the same software to develop their databases, some information on the structure and the data of the two databases is required since the title of an object or table might have different semantic for each database. Moreover, different tables may be developed to store the related data of an entity based on the analysis of the project. In order to solve this problem, first a knowledge base would be created for the entities and supplies of the databases and then an intelligent agent would be used for eliciting knowledge and justification process.

According to the current database technologies, usually the two relational and semantic structures are used for developing databases. The relational databases are based on mathematics and algebra rules and cannot develop a knowledge base for entities. In these groups of database systems, first the migration setting, including identification of related objects and tables, should be prepared using database scheme comparison devices [4, 5]. In the next phase, an expert checks the two databases semantically and in the last stage the migration process takes place using data migration tools [5, 6].

In this data migration process, the mapping phase is done by experts. This is time consuming and accompanied with error for large databases. The most important weakness of relational model in data migration is the lack of knowledge or semantic in the database structure and it is the reason for which semantic databases are presented [7]. Of the most obvious advantages of semantic databases compared to relational databases is the presence of semantic in the database structure. This makes possible the atomization of the first phase of data migration process. The purpose of this paper is to propose a model for migrating data between two relational and semantic databases. In this proposed model, the data migration process is done by developing a middle layer. In the rest of this paper, the traditional and semantic data migration process is discussed in Section 2 and then in Section 3 the architecture of the semantic data migration is explained. The structure of the proposed model for data migration is presented in Section 4. Finally, the conclusion of the article is given in Section 5.

2. DATA MIGRATION PROCESS

The data migration process includes two important phases. In the first phase, identification, elicitation and collation of entities are performed. In the second phase the data migration process between source entities and destination entities are carried out. Usually the existing data migration systems focus on the first phase. Thus the difference or merits of each of these systems depends on implementation of the first phase. With such view, the data migration systems can be classified into two general categories: traditional data migration systems and semantic data migration systems. In traditional system the first phase is conducted by the use of experts and the schema of the two databases [5, 8]. By using schema, similar entities are getting related, and with experts the relation between the rests of the entities is decided.

This approach is not applicable for large databases. Of the shortages of this approach are being time consuming and probability of making errors. Hence, there is a need for a software layer which can be replaced with experts in large projects. Among databases, the relational database [9] is the most popular one for modeling of which ER [10] is usually used. In order to create the software layer, expert's operation on identifying entities must be surveyed. The question is how an expert can identify two conceptually but yet namely different entities, while the machine can't do such a thing. The reason is what makes the expert different from machine; it is the knowledge base he utilizes. The expert has primary knowledge about the subject of the two databases, and by using semantic he identifies similar entities.

Therefore a knowledge base has to be created along with a database so that the software layer can carry out the first phase of data migration. Due to the variety of entities we cannot use a fixed knowledge base for all entities. Consequently, to create a knowledge base for a database structure [11] ontology [12, 13] is used. This is the reason for which the second category of data migration systems called semantic data migration is presented. In semantic data migration systems ontology is tried to be used in the first phase.

In the semantic data migration approach, requests are transferred to a semantic environment. This environment can be either between two semantic databases or a semantic layer on a relational database. In order to implement the semantic layer on a relational database, ontology of the relational database is needed. In the rest of this section the process of creating ontology and data migration in semantic database are reviewed. The semantic data migration is thoroughly explained in Section 3.

2.1 Creating Ontology for Relational Database

In order to create ontology, two approaches are used based on relational data model and XML. In this research, the data model IDEF1X [15] is selected among the other different data models for developing database and demonstrating entities and their relationships [16]. Ontology items including class, domain and attribute must be generated by the use of data model. The created ontology should cover the data model concepts so that it is useable for the software layer. Nevertheless, the ontology itself is a data model. The entities and relations by the use of ontology are demonstrated with classes. OWL_DL language is chosen as a proper syntax for demonstrating ontology [3]. There are many approaches that are used for the process of creating ontology, among which, DB2OWL is chosen [16]. In this method, entities are categorized according to relationship. First the parent and child tables are identified and converted to the super classes and sub classes. The rest of the tables are transferred into a class according to their role in the data model.

Despite the existence of data models such as SIM [17] and reverse engineering methods like using web pages [18] which try to inject semantic, there is no approach to express the semantic of each table and relation in a data model. Therefore the created ontology based on a data model is not semantically prefect. In the second approach, the database structure is first expressed in XML. The DTDs are the most important tools used for creating XML structures [19]. By the use of RDF, RDF schema, OWL_DL, and convert the XML structure to multiple semantic layers, a more accurate ontology compared to the relational data model can be created [3]. In the first stage, entity and relations are defined using RDF and RDF schema [3], and in the next stage the semantic relation between classes is created using the OWL_DL.

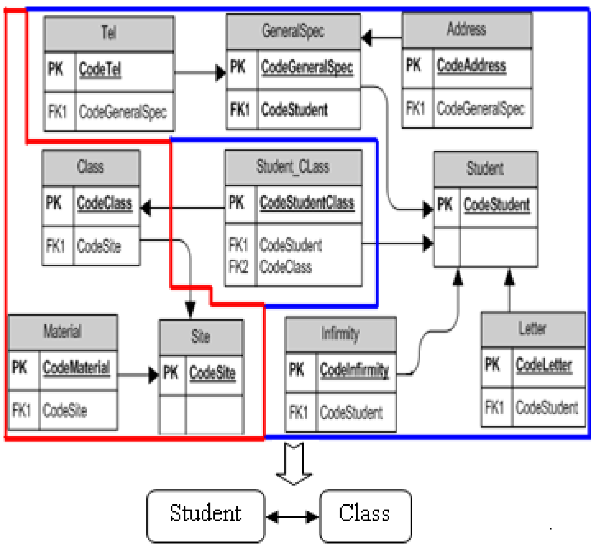

Fig 1: Create Ontology For Relational DataBase.

For example, consider a part of logical database design for an academic system demonstrated in Figure 1. In this system, the actual process is taking place between two entities: the student and the lecture. The student has several lectures in each semester and each of the lectures includes several students. This table is shown in the two zones in red and blue in Figure 1 respectively. The data concerning the student entity and lecture entity are stored in another table. The red zone or the blue one in Figure 1 shows this table.

By the use of the mentioned approaches, ontology is generated. In ontology generating process, the first thing that is taken care of is identifying parent and child tables. The child tables will be categorized under parent tables and the access to their data will be only granted via the parent table. A table which is between two parent tables will be assigned as their connective and it will be presented with a directional and labeled arrow in ontology charts.

2.2 Semantic Database Model

The existing database systems –both relational and object oriented– emphasize on data management and do not consider the semantic and semantic association between data and stored entities. This gives us a more abstract perception of the system [16]. The absence of a knowledge base and a standard in the stage of designing the entities makes the first phase of data migration difficult. In semantic database models (SDM), the definition of the entities and their relationship are done by the use of ontology [20]. Therefore the semantic database apart from managing data is able to store the semantic of data as some facts about the entities [7].

The presence of knowledge in definition of entities and their relationship makes semantic database closer to the real world model. Also it makes the requests easier to express and create semantic search engines to free the user from the complexity of relations between tables. These advantages in semantic database, makes the first phase of data migration easier and more applicable.

In this model, the focus is on gathering the semantic of data instead of relationships and attributes. The semantic model is richer than the traditional model, and can describe the fact found in data and the semantic relationship of objects better. In the semantic database all the operations including saving and retrieving data are based on semantics, however, the semantic database has more layers compared to the traditional model [21].

Figure 2 shows the architecture of semantic database. At the beginning, there is an application layer called API which its task is to get the low level instruction from the upper layer and carry it on to the semantic database. The next layer is the database engine. In fact this is where all semantic processes take place in order to carry out the data extraction. The user's request is submitted to the application layer and this layer generates the corresponding semantic query and submits it to the semantic engine. The semantic engine sends out the low level instruction to the application layer based on the semantic scheme created from the database.

For instance, the required queries for searching a student’s full information in the mentioned academic system in Section 2.1, using SQL, are presented in both relational and semantic approaches in Figures 3 and 4 respectively. In semantic approach, the user could write his request with a lower number of lines of codes and more importantly expressing his/her request much more simply. Also it is much closer to what he/she has in mind.

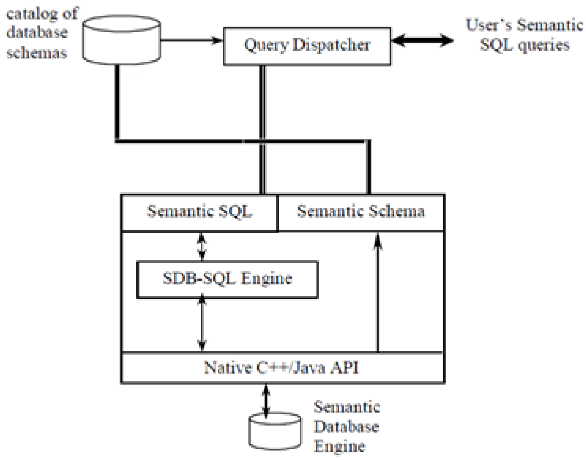

Fig 2: a Layer view of Semantic DataBase.

```
Select All_Information
From Student
Where  Code_Student = @CodeStudent
```

Fig 3: A Semantic Query.

The main data extraction operation is based on the existing semantic among database entities. Likewise, in data migration between two semantic databases, the process is based on semantic queries.

3. ARCHITECTURE OF SEMANTIC DATA MIGRATION MODELS

The required processes for using SDM technique for a relational database have been described in the previous Section. In the following Section, the architecture of each proposed model is introduced and reviewed.

3.1 The Merging of Ontology

This method includes two phases. In the first phase a semantic layer from two databases is created by ontology. In the second phase, classes are matched for data migration. A data model is created for each database for the first phase. Then a semantic model of the databases is created using one of the ontology creating methods [12, 13]. For the second stage, we need to create entities of classes. In fact a sample of classes is the one which is about to be migrated.

International Journal of Applied Information Systems (IJAIS) – ISSN : 2249-0868
Foundation of Computer Science FCS, New York, USA
Volume 3, No.3, July 2012 – www.ijais.org

```
WITH    SearchPhone ( [Type], [Phone] )
      AS (SELECT   [Type],
                    [Phone]
           FROM     General.Phone
           WHERE    CodeGeneralSpec = @CodeGeneralSpec
         )
SELECT  @TelHome = (SELECT Phone FROM   SearchPhone WHERE  [Type] = 'HOME'),
        @TelWork = (SELECT Phone FROM   SearchPhone WHERE  [Type] = 'WORK'),
        @Mobile = (SELECT  Phone FROM    SearchPhone WHERE   [Type] = 'MOBILE');
WITH    SearchEC ( EnrollmentDate, CodeCourse ,EnrollmentKind)
      AS (
           SELECT   TOP 1 CS.EnrollmentDate,
                    CC.CodeCourse ,CS.EnrollmentKind
           FROM     Class.StudentDetials AS CS
           INNER JOIN Class.Class AS CC ON CS.CodeClass = CC.CodeClass
           WHERE    CS.CodeStudent = @CodeStudent
                AND CS.EnrollmentDate = (SELECT MAX(EnrollmentDate)
                                          FROM Class.StudentDetials WHERE CodeStudent = @CodeStudent)
         )
SELECT  @EnrollmentDate = (SELECT  EnrollmentDate FROM    SearchEC),
        @EnrollmentKind = (SELECT  EnrollmentKind FROM    SearchEC) ,
        @CourseName = (SELECT CourseName FROM Course.Course WHERE   CodeCourse = (SELECT CodeCourse FROM SearchEC)) ;

WITH    Student ( CodeStudent, Name, Family, CountSession, Discount )
      AS (
           SELECT   SS.CodeStudent,
                    SS.Name,
                    SS.Family,
                    SS.CountSession,
                    SS.Discount
           FROM     Student.Student AS SS
           WHERE    SS.CodeStudent = @CodeStudent
         )
SELECT  SS.CodeStudent,
        @EnrollmentKind AS 'EnrollmentKind',SS.Name,SS.Family,SS.CountSession,SS.Discount,GG.BirthdayDate,GG.Email,
        GG.ExportPlace,GG.F_Name,GG.Filament,CASE GG.Gender WHEN 'TRUE' THEN 'Male' WHEN 'FALSE' THEN 'Female' END,
        GG.IdentityCard,GG.Picture,GG.PostCode,GG.SSN,GG.Testimony,GG.Job,@TelHome AS 'TelHome',@TelWork AS 'TelWork',
        @Mobile AS 'Mobile',@AddressHome AS 'AddressHome',@AddressWork AS 'AddressWork',@CourseName AS 'CourseName',
        @EnrollmentDate AS 'EnrollmentDate',SI.KindInfirmity,SI.CLM2,SI.CLM3,SI.CLM4
FROM    Student AS SS
INNER JOIN General.GeneralSpec AS GG ON SS.CodeStudent = GG.CodeObject
LEFT OUTER JOIN Student.Infirmity AS SI ON SS.CodeStudent = SI.CodeInfirmity
WHERE   GG.[Date] = (SELECT MAX([Date]) FROM  General.GeneralSpec  WHERE  CodeObject = SS.CodeStudent)
```

Fig 4: a Relational Query with SQL Server 2008.

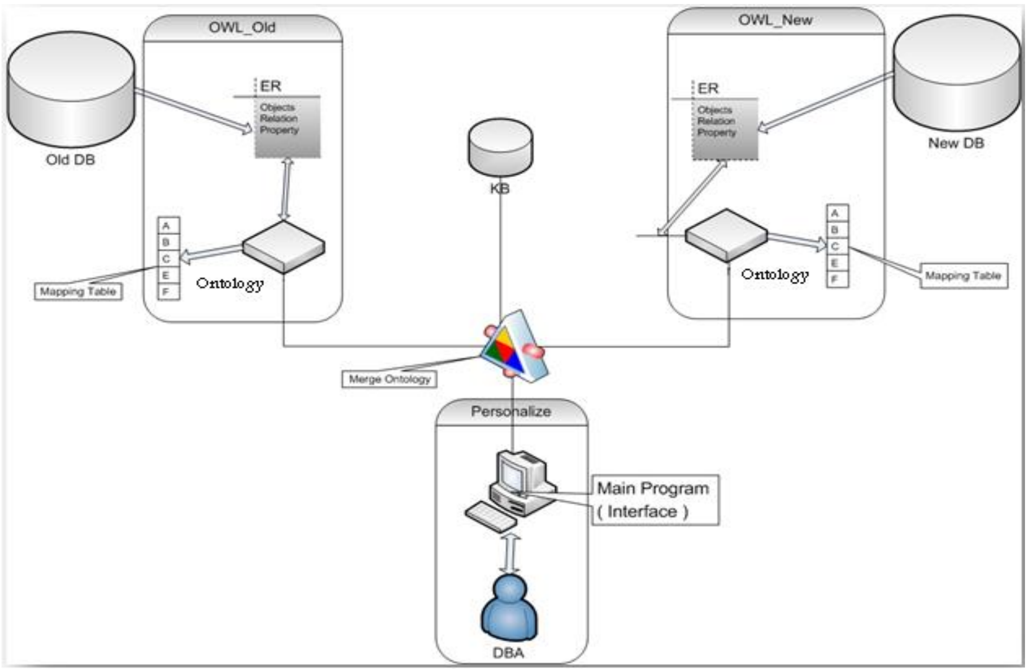

Fig 5: Architecture of Merge Ontology Model for Semantic Data Migration.

It is needed to divide the migration process into three stages to migrate data of entities [22]. First a data scheme is created. Then the created scheme gets converted to a semantic scheme using ontology. The data migration process to class is carried out according to predefined rules. Figure 5 shows an overview of this model. According to the mentioned stages, it is necessary to have a knowledge base for ontology merging. This knowledge base is created in two ways. The first method is manual. User, imports in advance the mapping table to integrate and associate the classes' name. The second method does this automatically using semantic data model.

3.2 Semantic Convertor (Middle Ware)

The goal of ontology merging model is to completely transfer data from the source database to the destination database. In such cases, either both of databases are semantic or one semantic layer is created for them. In some cases, there might be no need to transfer or entirely upgrade the existing (usually a relational) database. In the mentioned state, the database continues carrying on its job and the new database or application sends its request to this database in special cases.

Semantic convertor is proposed to solve this issue. The idea behind semantic convertor is to create a software layer for relational database [21]. An illustration of this model can be seen in Figure 6. The task of this layer is to create a semantic scheme from the database which will convert the semantic SQL instructions to their counterpart relational SQL instructions.

The user can send request semantically without altering any changes in the databases, and then receives the response relationally.

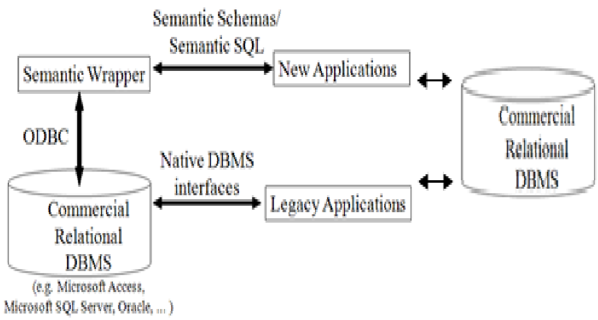

Fig 6: Architecture of Semantic Middle Ware.

4. PROPOSED DATA MIGRATION MODEL

Based on the studies conducted, the proposed models are usually applicable for data migration between two semantic databases or two traditional one. According to the new trend, in developing semantic database, the data migration between two semantic and traditional databases gets special importance.

Hence, in this paper, a data migration model is proposed between two semantic and traditional databases. The main focus of the proposed model is on the relation between the two traditional and semantic databases. In this case the semantic database is assumed as the source and the traditional database as the destination. In this method a middle layer is placed between two databases. This middle layer consists of two phase of creating an entity and filling it. Figure 7 shows an overview of this model.

In the first phase, the user’s semantic request is converted into an empty entity. In the next phase, the middle layer will fill these entities using the data from the existing relational database. Respondent to the Figure 8, the user’s request is syntactically checked at first. Then the user’s requests tried to be matched using class templates, and descriptive and syntactic texts found on the arrows between classes.

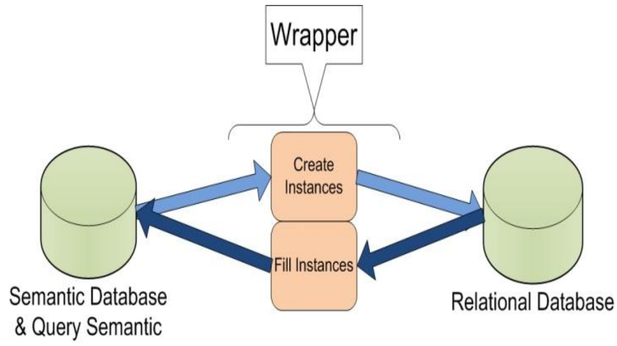

Fig 7: View of Proposed Data Migration Model.

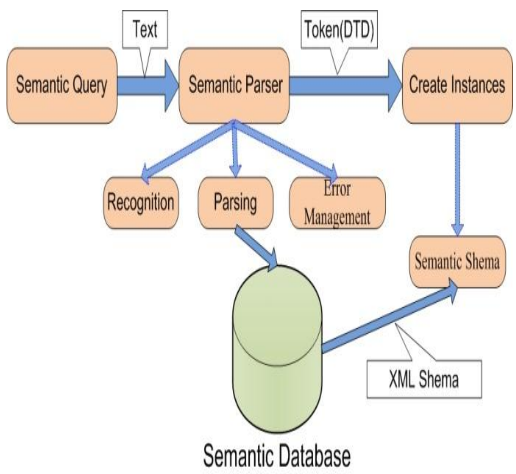

Fig 8: Create Instances.

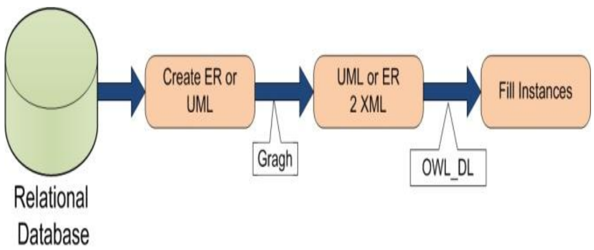

Fig 9: Fill Instances.

By going through this process, the required entities are created using XML. By the help of these prototypes, and the semantic scheme of the semantic database, several instances are proposed. In the second phase, the entities must be filled. In order to do that, the data are presented in the XML form using the relational database structure [19, 24]. Then a semantic scheme is created from the data by using ontology [25]. By creating the ontology, the mapping operation between entities and ontology is carried out.

After mapping, the filling of the entities is carried out. Figure 9 demonstrates these phases. In this phase the existence of several instances, makes it possible to choose at least one appropriate instance. An instance of this process is shown in the mentioned example in Section 2.1. This layer, maps the semantic request presented in Figure 3 to the relational request shown in Figure 4. From the final user's perspective, this method utilizes the encapsulation technique for data migration process, and thus the user is not involved in the complexity of carrying on the task of migrating data. The data migration request from different data in relational database, and its applicability for most of data models are the important advantages of this migration method.

5. CONCLUSION

In this article the methods of data migration between databases have been studied. Regardless of the technologies employed, the common point among all these databases is the necessity of a knowledge base for mapping between two databases. The knowledge base is used for extracting the structure and mapping the entities of the databases. The structure of semantic database makes the creation of a knowledge base easy. But, generally creating a knowledge base for a relational database is a difficult task. Having this in mind, in the proposed model, based on a semantic model, a semantic layer is created by which the transaction is carried out. The middle layer by using the semantic database ontology gets the needed knowledge for mapping between requests. One of the important applications of this method is implementing the multilingual search engines. The request is sent to the source language database and the search engine, searches among databases.

6. REFERENCES

[1] Data_migration, http://en.wikipedia.org/ wiki/ Data_, accessed 2010.

[2] Hugill, A. and Zhou, H. 2007. Data Interoperability via Ontology-Based Data Understanding. http://www.mti. dmu. ac.uk.

[3] Antoniou, G. and Harmelen, F. V. 2008. A Semantic Web Primer. ISBN-13:978-0-262-01242-3.

[4] "Apex SQL Diff", Apex SQL. 2010. http://www.apexsql.com /sql_tools_diff.aspx.

[5] "Red Gate SQL Compare", http://www.red-gate.com/ products/SQL_Compare/index. htm, accessed 2010.

[6] "Apex SQL Data Diff", http://www.apexsql.com/ sql_tools_datadiff.aspx, accessed 2010.

[7] Rishe, N. ODB: Semantic Object DBMS, High performance Database Research Center School of Computer Science University Park, ECS243, Miami, FL3319, http://hpdrc.fiu.edu/.

[8] "Apex SQL Comparison Studio", http://www.apexsql.com/sql_tools_comparison_studio.aspx, accessed 2010.

[9] Elmasri, R. and Navathe, S. B. 2004. Fundamentals of DataBase Systems, Pearson Education, ISBN 0-321-12226-7, Fourth Edition.

[10] Chen, P. P. 2002. Entity-Relationship Modelling: Historical Events, Future Trends, and Lessons Learned, Springer- Verlag New York, Inc, 296-310.

[11] Saias, J. and Quaresma, P. 2003. A Methodology to Create Ontology-Based Information Retrieval Systems. Computer Science, Vol. 2902/2003, 424-434.

[12] Noy, N. F. and McGuinness, D. L. 2001. Ontology Development 101: A Guide to Creating Your First Ontology. Stanford Knowledge Systems Laboratory Technical Report KSL-01-05 and Stanford Medical Informatics Technical Report SMI-2001-0880.

[13] Liu, L. and TamerOzsu, M. 2008. Ontology. Entry in the Encyclopedia of Database Systems, Springer-Verlag.

[14] Tujarov, H. and Mihailov, I. 2004. ER model: semantic data models developing. ACM, ISBN: 954-9641-38-4.

[15] Davidson, L. 2008. Pro SQL Server 2008 Relational Database Design and Implementation. Apress, ISBN-13 (electronic): 978-1-4302-0867-9.

[16] Cullot, N., Ghawi, R. and Yetongnon, K. 2007. B2OWL: A Tool for Automatic Database to Ontology Mapping. In Proc. Of 15th Italian Symposium on Advanced Database Systems (SEBD2007), 491-494,TorreCanne, Italy.

[17] Fritchman, B. L., Guck, R. L., Jagannathan, D., Thompson, J. P. and Tolbret, D. M. 1989. SIM: design and implementation of a semantic database system. in:Proceedings of The ACM SIGMOD Conference, 46–55.

[18] Premerlani, W. and Blaha, M. 1994. An Approach for Reverse Engineering of Relational Databases. Communications of the ACM, Vol.37, No.5, 42–49.

[19] Ma, Z. M. and Yan, L. 2007. Fuzzy XML data modeling with the UML and relational data models. Data & Knowledge Engineering-DKE, Vol.63, No.3, 972-996.

[20] Hammer, M. and McLeod, D. 1981. Database Description with DM: A Semantic Database Model. ACM Transactions on Database Systems- TODS, Vol.6, No.3, 351-386.

[21] Rishe, N. Yuan, J. Athauda, R. Chen, S. Lu, X. Ma, X. Vaschillo, A. Shaposhnikov, A. and Vasilevsky. D. 2000. Semantic Access: Semantic Interface for Querying Databases. In Proc. Of the 26th VLDB, 591–594.

[22] Wimmer, M. 2009. A Meta – Framework for Generating Ontologies from Legacy Schemas. dexa, 474-479, 20th International Workshop on Database and Expert Systems Application.

[23] Hitzler, P. Krotzsch, M. Ehrig, M.and Sure, Y. 2005. What is Ontology Merging? American Association for Artificial Intelligence.

[24] Marchewka, J. T. and Liu, C. 2002. Mapping UML to XML: A Method for Teaching an Object-Oriented Approach to E-Business Development. Journal of Informatics Education and Research, 27-34.

[25] Rodrigues, T. Rosa, P. and Cardoso, J. 2006. Mapping XML to Existing OWL Ontologies. International Conference WWW/Internet.